

Fiche SmartGit

Contenu

Installer SmartGit sous Ubuntu	2
Installer SmartGit sous Windows	2
Configuration de SmartGit	2
Vue d'ensemble de SmartGit	2
Créer une branche.....	3
Faire un commit avec SmartGit.....	3
Tirer les sources.....	4
Rebase	4
Régler les conflits	4

Toutes les commandes git qui ont été vues dans la fiche github peuvent se faire faire l'aide d'outils graphiques dédiés à Git. Ici, nous nous proposons de présenter l'outil multiplateformes SmartGit.

Installer SmartGit sous Ubuntu

<http://www.kvcodes.com/2016/03/install-uninstall-smartgit-ubuntu/>

Installer SmartGit sous Windows

<http://www.syntevo.com/smartgit/download>

Configuration de SmartGit

Pour ouvrir un répertoire dans SmartGit, il suffit de cliquer sur Repository > Add or Create, puis de choisir le bon répertoire.

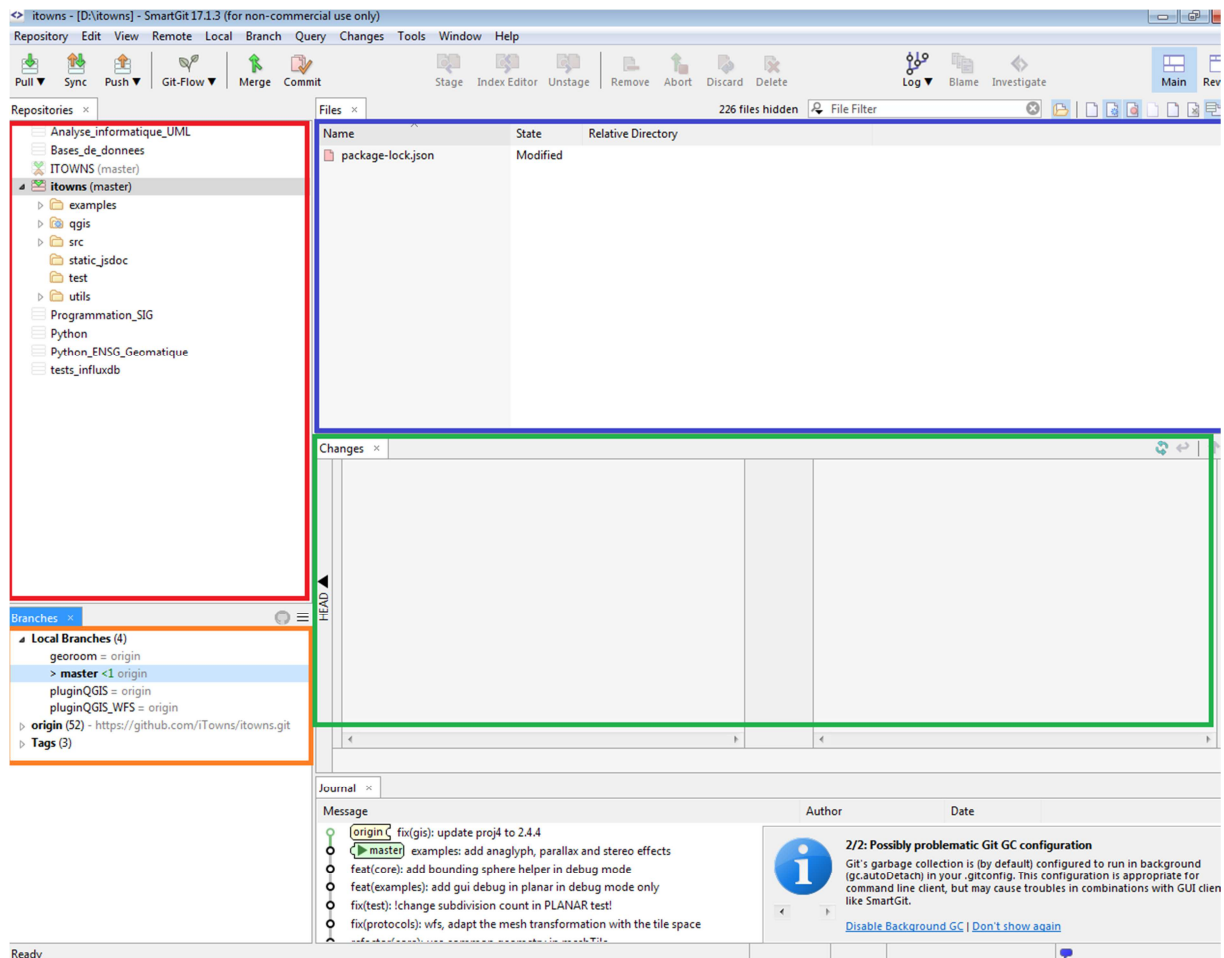
Vous pouvez aussi cloner directement un dépôt Github : Repository > Clone... et renseigner l'adresse du dépôt avec le bon protocole (HTTPS ou SSH).

Il sera peut être nécessaire de configurer le proxy dans smartgit pour cette étape

Vue d'ensemble de SmartGit

Sur la vue ci-dessous :

- En rouge, le (ou les) répertoire(s) de travail
- En bleu, les fichiers qui ont été modifiés
- En vert, les différences qui ont été apportées sur un fichier (fichier sélectionné dans la zone bleue)
- En orange, les branches qui existent en local et sur le serveur.



Créer une branche

Pour créer une branche depuis une branche existante sur le serveur github, il suffit de la trouver en cliquant sur « origin » dans la partie orange et double cliquer sur la branche voulue. SmartGit va alors vous proposer de créer une branche en locale portant le même nom.

Vous pouvez aussi créer une nouvelle branche qui n'existe pas, dans le menu 'Branch', cliquer sur 'Add Branch...' : SmartGit va alors vous demander de nommer votre nouvelle branche.

Attention : dans le deuxième cas, SmartGit va créer une branche à partir de celle sur laquelle vous êtes au moment de la création, donc si vous voulez créer une branche identique à la branche master en vue de la modifier, placez-vous au préalable sur la branche master.

Faire un commit avec SmartGit

Dans la partie bleue : sélectionnez les fichiers que vous voulez commiter, puis cliquer sur commit, SmartGit vous propose alors de renseigner le message du commit.

Tirer les sources

Il suffit de cliquer sur le bouton « Pull » en haut à gauche, smartgit vous propose plusieurs options, choisissez 'Fetch only' (il va juste tirer les sources sans les fusionner avec votre branche).

Rebase

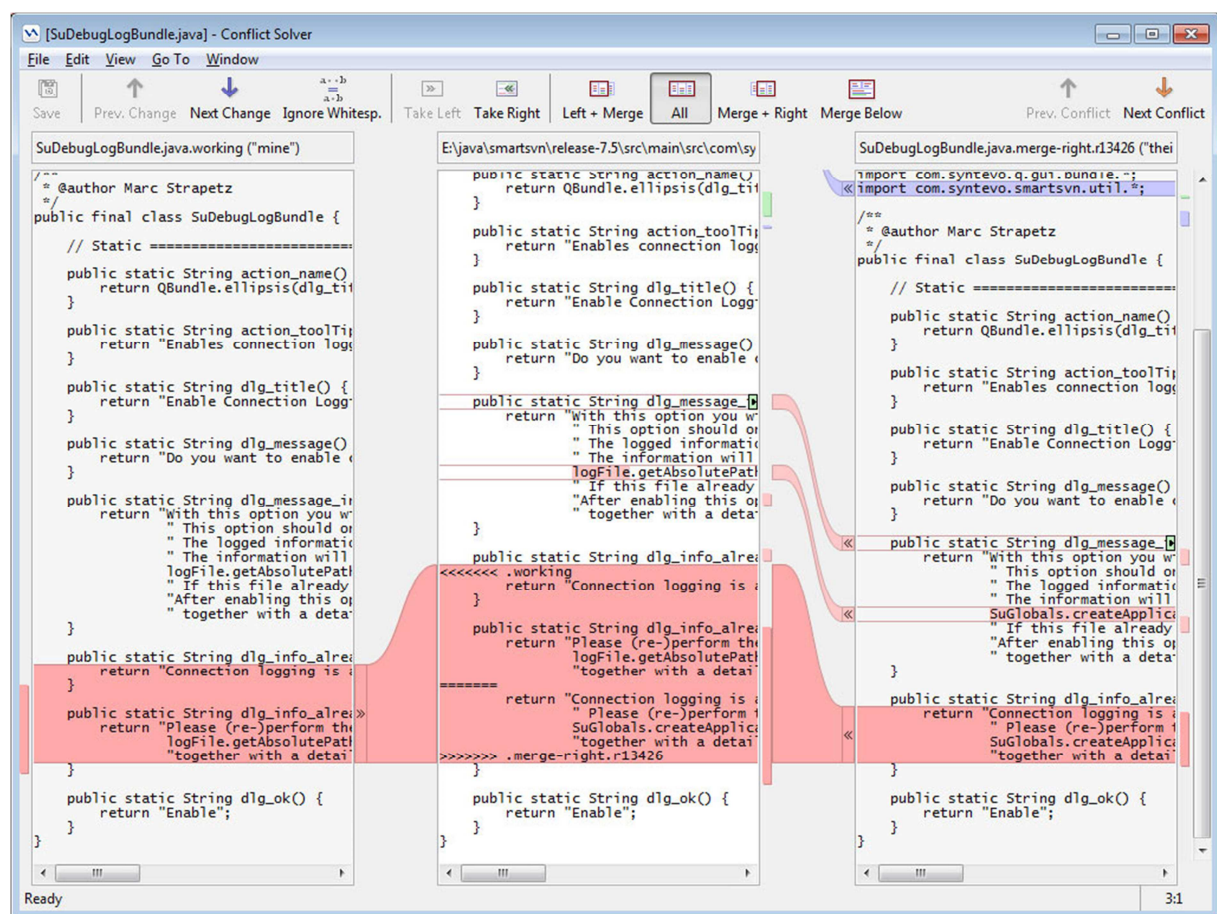
Personnellement je vous conseille de le faire en ligne de commande, il est possible d'avoir des surprises avec SmartGit.

Par contre Smartgit est très efficace pour régler les conflits qu'il peut y avoir lors d'un rebase.

Régler les conflits

Si Git vous signale des conflits lors d'un rebase, je vous conseille de repasser sur Smartgit, plutôt que de les régler avec un éditeur de texte...

Dans la partie bleue, smartgit vous signale les fichiers qui sont en conflits. Faites un clic droit sur l'un de ses fichiers puis cliquez sur 'Conflict Solver'. SmartGit vous propose alors une vue avec 3 éditeurs de texte.



A gauche, votre code tel qu'il est sur votre branche ('mine' ou 'yours').

A droite, le code de la branche depuis laquelle on se base pour faire notre rebase ('theirs').

Au centre, le code avec les conflits : c'est là que vous devez agir ('working tree').

SmartGit vous signale les conflits en les surlignant en rouge, il est alors à vous de juger ce qu'il est bon de garder dans l'éditeur central. Une fois que vous avez fini, n'oubliez pas d'enregistrer et de fermer le conflict solver (smartgit vous proposera alors de 'stasher' vos fichiers, ce qu'il faut faire !).

Une fois tous les conflits résolus, vous pouvez cliquer sur le bouton « continue » dans l'interface de smartgit ou de lancer la commande

➤ **Git rebase --continue**